

## REMARKS

Reconsideration of this application as amended is respectfully requested.

In the Office Action dated September 11, 2008, claims 1-29 are pending. Claims 1-29 stand rejected. In this response, claims 1, 8, 15, 17, 23, 29 have been amended. No new claims have been added. No claims have been canceled. Thus, claims 1-29 remain pending. Support for the amendments can be found throughout the specifications as filed. No new matter has been added.

### *Rejections under 35 U.S.C. § 102(b)*

#### Claims 1-2, 8-9 and 15-16

Claims 1-2, 8-9 and 15-16 stand rejected under 35 U.S.C. §102(b) as being anticipated by “Tolerating Memory Latency through Software-Controlled Pre-Execution in Simultaneous Multithreading Processors” by Luk (hereinafter “Luk”). However, applicants respectfully submit that applicants’ claims 1-2, 8-9 and 15-16, as amended, are not anticipated by the cited reference.

Specifically, for example, independent claim 1 includes the limitations:

“analyzing source codes of a main thread having one or more delinquent loads, the one or more delinquent loads representing loads which likely suffer cache misses during an execution of the main thread, the source codes including one or more code regions, each code region corresponding to a sequence of instructions representing an iteration loop in the source codes, the one or more code regions sharing at least one instruction in the source codes;

selecting a code region from the one or more code regions for one or more helper threads with respect to the main thread based on the analysis; and

generating codes for the one or more helper threads, the one or more helper threads being speculatively executed in parallel with the main thread to perform one or more tasks for the region of the main thread”

(emphasis added)

Applicants submit that Luk fails to teach or suggest the limitation of selecting a code region from one or more code regions representing iteration loops sharing instructions in source codes of a main thread based on an analysis of the source codes to generate one or more helper threads.

Rather, Luk provides software-controlled pre-execution schemes to help programmers writing memory-friendly programs using pre-execution threads and for compiler to insert pre-execution (Luk, page 40, col. 2, par. 4 – page 41, col. 1, par. 1). Luk proposes ISA (Instruction Set Architecture) extension with three new instructions to spawn a pre-execution thread at a particular PC (Program Counter) and stops when certain number of instructions have been executed in the pre-execution (Luk, page 44, col. 1, pars. 4-5, col. 1, par. 1). According to Luk, a compiler inserts pre-execution by determining which references could benefit from the pre-execution and calculating the pre-execution distance (Luk, page 44, col. 2, par. 4). In addition, Luk discusses performing locality analysis to locate where cache misses occur in a program (Luk, page 51, col 2, par. Phase 1, Step 1). Specifically, Luk demonstrates code fragment tmp->next for determining data access for locality analysis (Luk, page 51, col 2, par. Phase 1, Step 3). Thus, Luk's analysis identifies which data access instruction in a program would benefit from a thread pre-execution. Clearly, a single data access instruction cannot logically represent a code region for iteration loops. Nowhere does Luk teach or suggest selecting a code region from one or more code regions representing iteration loops sharing instructions in source codes of a main thread.

The Office Actions asserts that:

“In addition to identifying the data access instruction ... Luk discloses identifying the source code which would generate the necessary address (e.g. ... calculate the pre-existence distance ... discover multiple paths and procedure calls) ...”  
(Office Action, page 2)

It appears that the Office Action alleges Luk discloses a code region of source codes as a data access instruction. Applicants respectfully disagree with the Office Action. Here, Luk discusses using compilers to recognize pointer-based data structure and array accesses (Luk, Appendix, Phase 1, Step 2, page 44, col. 2, par. 3). It is respectfully submitted that one with ordinary skill in the art would not recognize a code region of source codes based on Luk's teaching.

In order to anticipate a claim, each and every limitation of the claim must be taught by the cited reference. It is respectfully submitted that Luk fails to disclose the limitations set forth above. Therefore, it is respectfully submitted that independent claim 1, as amended, is not anticipated by Luk.

Independent claims 8 and 15, as amended, include similar limitations as noted above. Therefore, for at least the similar reasons as discussed above, it is respectfully submitted that claims 8 and 15, as amended, are not anticipated by Luk.

Given that claims 2, 9 and 16, as amended, depend from and include all limitations of one of independent claims 1, 8 and 15, as amended, applicants respectfully submit that claims 2, 9 and 16, as amended, are not anticipated by Luk.

***Rejections Under 35 U.S.C. 103(a)***

**Claims 3-4, 10-11, 5-7, 12-14, 17-19, 22-25, 28-29, 20-21, 26-27**

Claims 3-4 and 10-11 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Luk in view of “Exploiting hardware Performance Counters with Flow and Context Sensitive Profiling” by Ammons et al. (hereinafter “Ammons”). Claims 5-7, 2(12)-14, 17-19, 22-25 and 28-29 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Luk in view of “Data Prefetching by Dependence Graph Precomputation” by Annavararam et al. (hereinafter “Annavararam”). Claims 20-21 and 26-27 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Luk in view of US Patent No. 7,243,267 to Kelmm et al. (hereinafter “Klemm”). Applicants hereby reserve the right to swear behind Kelmm at a later date. However, applicants respectfully submit that applicants’ claims 3-4, 10-11, 5-7, 12-14, 17-19, 22-25, 28-29, 20-21 and 26-27, as amended, are patentable over the cited references.

These claims depend from one of the above independent claims. It is respectfully submitted that none of the cited references herein, individually or in combination, disclose or suggest the limitations set forth above. Furthermore, there are neither motivations nor suggestions to combine the cited references. Therefore, for the reasons similar to those discussed above, these claims are patentable over the cited references.

## CONCLUSION

In view of the foregoing, it is respectfully submitted that the applicable rejections and objections have been overcome.

Authorization is now given to charge our Deposit Account No. 02-2666 for any charges that may be due. Furthermore, if an extension is required, then applicant hereby requests such an extension.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP

Date: May 20, 2009

/Vincent WenJeng Lue, Reg. No. 56,564/  
Vincent\_Lue@bstz.com  
Vincent WenJeng Lue  
Reg. No. 56,564

1279 Oakmead Parkway  
Sunnyvale, California 94086-4040  
(408) 720-8300